EXHIBIT A

RECEIVED
CENTRAL FAX CENTER
DEC 23 2004

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

In re Application of: Steve Hole
Serial No.: 09/765,927
Filed: January 19, 2001
For: Message Tracking System and Method

Art Unit: 2144
Examiner: Brown, James Lee

Commissioner for Patents
Washington, D.C. 20231

Declaration Under 37 C.F.R. §1.131

Steve Hole, a resident of Edmonton, Canada, declares as follows:

1. I am the inventor of the subject matter claimed in the above-identified patent application.

2. I conceived and reduced to practice in Canada the invention claimed in the above-identified patent application prior to April 14, 2000, the earliest filing date of the cited Unites States Patent Application Publication No. 2001/0042131 to Mathon et al ("Mathon et al.").

3. Attached as Exhibit A is a copy of sample pages from the computer source code that constitutes a reduction to practice of the invention claimed in the above-identified patent application prior to the filing date of Mathon et al. Exhibit A includes commentary pages from that computer source code. For example, page 1 describes "M-Builder" that includes a set of tools and libraries, which implement the invention claimed in the above-identified patent application. Similarly, page 2 explains the "Transaction Message Reception API," and page 4 explains the "Message tracking subsystem" in that computer source code.

SH.

4. That computer source code was completed after December 8, 1993 in Canada. Canada is a NAFTA and WTO country.

5. Exhibit A, which relates to the aforementioned conception and actual reduction to practice prior to the filing date of the Mathon et al., corresponds to the invention broadly disclosed and claimed in the above-identified patent application.

6. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: 21 Dec 04 Steve Hole

Steve Hole

M-Builder is a set of tools and libraries for creating applications that deliver complex data over the Internet. Data is transmitted using "transaction messaging" that is:

* Secure - message content (payload) can be protected, verified and authenticated for and during transmission.
* Reliable - message delivery or non-delivery can be guaranteed to specific addresses.
* Traceable - the status of a particular message can be determined at any time.

Legacy applications are extended to communicate with remote applications or users by integrating the M-Builder transaction messaging components. M-Builder includes a set of easy to use programmatic interfaces for:

* Creating and submitting messages of arbitrary complexity and payload type.
* Requesting various levels of security, reliability and traceability on messages.
* Receiving and extracting data from messages of arbitrary complexity and payload type.
* Automatically handling security and tracking requirements on incoming messages.

M-Builder interfaces are supported on the majority of common platforms including Unix, Windows and Macintosh, and in a number of common high and low level programming languages including C/C++, Tcl/Tk, and Perl.

4. Transaction Message Reception API

Feature Tasks	Description	Cost	D-Date
Recieve mailbox monitor * monitor setup/teardown API * mailbox connector modules * message decomposition modules * notification dispatch manager * module test harness * integrate into platform suite	Application library that implements a mailbox monitor. A mailbox monitor connects to and monitors mailboxes for incoming messages, pulls the messages from the mailbox, decomposes the messages (including performing all security operations) and then initiates event notifications for all application components that have registered an interest in messages of this type.	1 d 3 d 3 d 5 d 5 d 10 d	
C/C++ event manager API module receive * C event notification interface * C++ wrapper class * integrate into platform suite * module test harnesses	C/C++ language implementation of the event API. These also make up the core modules for the scripting level languages Tcl and Perl. Receive events occur when a mailbox monitor notifies an application that a message with a specific set of characteristics (MIME type, originator address etc.) arrives in one of the monitored mailboxes. This API defines the interface by which notifications can be made to an application. It includes registration and callback interfaces.	3 d 1 d 2 d 1 d	
Tcl/Tk event manager API module manager API. * implement Tcl extension module * integrate into platform suite * module test harnesses	Tcl language implementation of the event	3 d 2 d 3 d	
Perl event manager API module manager * implement Perl extension module API. * integrate into platform suite * module test harnesses	Perl language implementation of the event	3 d 2 d 2 d	
Java event manager API module manager * implement native Java class API. * integrate into platform suite * module test harnesses	Java language implementation of the event	5 d 10 d 5 d	

Transaction reception API user guide User guide describing how API can be used, how applications are constructed and what tools are available with the toolkit.

5. Message tracking subsystem

Feature Tasks	Description	Cost	D-Date
Message tracking database status.	Database for managing submitted message status. This includes the current status of the message, details about error conditions (if any) and message transition timing information.		
* define database schema		4 h	
* database management API		3 d	
* module test harnesses		1 d	
Message tracking registration	Register outbound messages in the tracking database. This enables tracking status to be updated and reports generated for a messages.		
* extend message submission API to register		1 d	
Message tracking monitor tracking	Tracking monitor collects asynchronous notifications from mailboxes and automatically updates message status in the tracking database.		
* DSN message handler		2 d	
* MDN message handler		2 d	
* tracking database status update		1 d	
* monitor daemon application		2 d	
Message tracking agent API	API for executing status queries against the tracking database on behalf of an authorized user.		
Message tracking report generator Reports	Generate reports based on business rules. can include alerts for certain kinds of failures. Reports are generated based on criteria like: bounced (rejected) messages, delayed (in transit) messages, lost messages.		
Message tracking agent GUI messages	UI for querying the status of one or more directly from the database. Includes support for initiating report generation.		